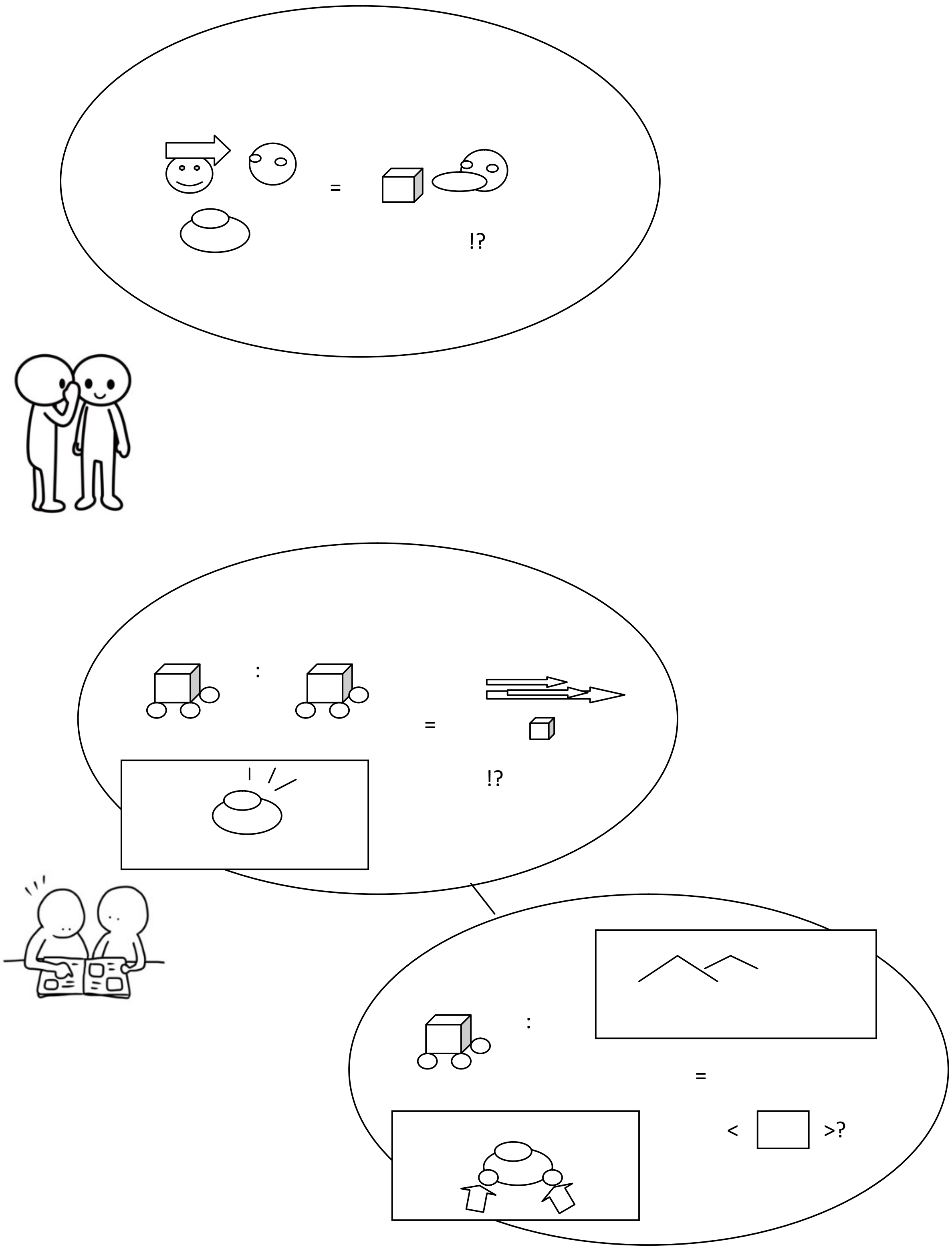
GraphSpeak



What is graphspeak

Graphspeak is an experimental method of communication

Where the transmission medium is based on <permutative> Graphs

By permutative, we mean that it requires a

Pre-handshake; selection of sampling nodes

That is likely to be confirm-able in <known time>

transmutes [handshake] transmute

transmutes aha! vectors are

vectors

Graphspeak is an attempt at creating a baseline of

Several common transcriptive modifiers to known concepts

By first decomposing

Starting with common protocols like

Graph Language @ Graph Theory

(#Bronstein, et al. 2021) on Geometric Deep Learning

However; eventually the notion of these language

Being a successful medium, does not actually depend on their RIGOR

But rather on their rating of <empathy>

Which is a middle layer of

< simulated handshake >   already commonly understood

Between humans, this means that ,

On the contrary to rigor and re-producibility

.

It actually depends instead on the notion of

Composition between   < the recipient's capability x the recipient's context >

By context we mean   time allotted,

amount of <n> (expected repetibility based on mood states / et)

By capability we mean   tools shared,

this amounts to limbs; or a middle layer

##

for a emulated handshake

.

Is based on these 3 sweeping process:

- Sampling original intention and output using Graph Theory (math)
- Converting it based on a staple table of known tools (limbs, etc)
- Decomposing the Graph Theory nodes onto connected middle layers
- Representing the handshake

In summary; sampling the original transformation; then decomposing it down, Onto hoping to figure out types of handshakes

Which are TOOLS / CONTEXT

By comparing the decomposed nodes for intersections

.

Our applied transformations possibly are:

Utilizing Prompting and

LLM output for

Layer 0	using Graph Theory to sample and position the nodes	
First conversion	per usual permutative rules adopted (below)	
seperate		

Layer 1	converting them using chosen (currently only 3)	
First conversion	permutative format (and accepted tools)	(see p8+)
Separate	GraphT -> GraphF + t	
	(F representing our chosen medium for now)	

Layer 2	identifying the agentic / recipient tools and toolkit	
Ontop of layer1	then decomposing both GraphT <> GraphF	
Continuation	upon decomposition;	
	then finding out the % of similarity between <> rigor v tool	
	then trying to lower the range for optimal form	

Our objective is to:

1. Test its deployment for kids

(results may depend more on how fun the game is; still!)

2. Test its deployment for cross-Language purposes!

Generalizability among languages

3. Test its deployment for RIGOR

Tho this was motivated by permutative- handshake;

given that we will find out the details on which these ratios are obtained; it is possible that given time;

we could figure out why certain things are missing a rigor or requiring a rigor; or acceptable in terms of rigor for practical purposes

which might have downstream functions for other tools

.

About: Emulated Handshake

Our perspective on this is that; since this is a (possibly; likely)

a ever-changing trend-based , circumstance based,

notion of carried context by participants in a shared culture

ferried or intelligently-bet upon the success of transmission by [x] amount of quick iteration.

We should instead focus on mining these and perceiving this as

An Upper Layer

[ ] of graph

To then decompose further down using the transmutations

[] [] []

[][][] then between a number of these decomposed forests

we might spot what ends up being a cross-similarity

that represent those *handshakes

Then seeing if the transmutations

Based on

1. Tool keeping Or
2. Shared context

Would eventually come down to similar notions

(in this case; perhaps about LIMB or

more evolved from such; Mobility

Some shared perspective on theory of mind on why this is likely

A good approach; are such like:

Paper

Self-other overlap (Shah et al. 2023; Askell et al. 2021; 2023) on Deployment

.

Along with the core theory representing the process and effort

We also seek to make a game;

On which seeks to picture itself this way

“ Please Help this ## person , compose a Graph “

as an interactive way to both test & apply

graphspeak

For such game, we likely would do these

- Provide a bevy of graphs 3-3 node connections that are
- Pre-selected for fun / lighthearted purposes
- (Which does help in the function of “Handshake”)

- We then prompt the player to compose a tree of graphs
  To achieve their goal (usually engineering related)

- On top of that; subsequent episodes require them to apply
- Our preselected transmutation types to the graphs
- Which are detailed below
- (referred to before as (chosen) Permutative formats)

- Proceed to the decomposition and other game features
- to query for handshake or shared notions

Selection table for first permutation

(will update later on; currently only 3 for starting small)

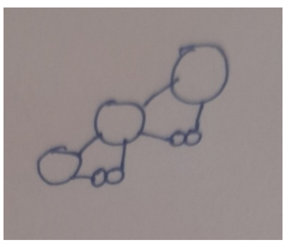
(we will have a permutative table; and a toolkit table)

Meanwhile; here are the current (temp) norms

(1) Node1 -(/+)Node2 -(/+)Node3 = Node4 -> Node1 –oo –Node2 –oo –Node3 =

(2) Node1 + Node2 = Node 3 | Node1b +(~) Node2b = Node 3/3b

(3) Node1 + Node2 = Node3 -> Node1[1a.1b.1c / 1a.oo.1b.oo] + Node2[2a.2b] = Node3



Tr1

(1)

Node1 + (or <->) Node2 +(/) Node3 = Node4

>

Node1 plus or to Node2 plus or to Node3 becomes Node 4

Converted onto

Node1 oo Node2 oo Node3 = Node4

Which

| oo | denotes any other transmutative methods in between

i.e. Engine. Wheel. Chasis = Car

onto Engine –[snug-onto][snug-chasis]- Chasis - [snug-onto][snug-chasis] – Wheel = Car side note that;

In this permutative example

we don’t have to prioritize accuracy

It is expected that each of these permutations are only

above nn% correct

(arbitrary middle range number (50%?))

Our approach first hope to gather base intuition from LLM’s

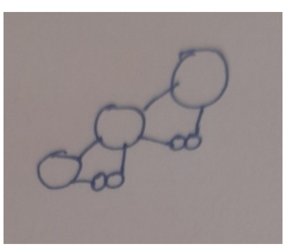
- Adding a human labeling layer
- Hopefully to able to generalize
- And port some of these process
- Also to a LLM Query

Before confirming them. decomposing them

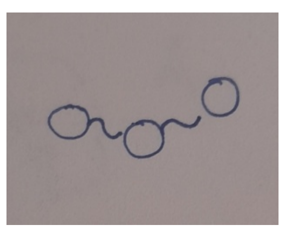
And finding out whether theres anything in between

That is good to generalize again and hand it back to

Combine & compare



[oo] :



[~] ([ ])

Say starting with assigning similarity %

Which could be workable (spot the similarity between disparity of %)

.

Example on T1

Ex: “Hey claude, between these 3 main process or component; what is done between them?”

“Hey claude; do you know what they usually do along with these 3 listed things?”

et

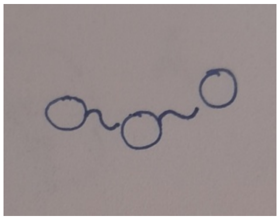
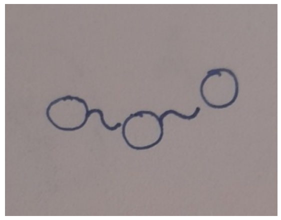
Ex2: “Hey claude; given that this has been done;

Why would this person give this analogy? Can you identify the process

“ and what would be akin of?

T2

pic

 |  Tr2

(2)

Node1 + Node2 = Node 3 | Node1b + Node2b = Node3/3b

Node1 ~ Node2 ~=(?) Node 3 | Node1b ~ Node2b ~=(?) Node3 /3b

-

in words: CHASIS plug to WHEEL [to CAR ..]

as if LEGS plug to BODY [to MAN..]

then decompose each to layers of tools

(or even lower)

In any case; the symbol of ~

(means that it can be anything; it can be filled with anything and any stretch

But an agent (operative agent) should be build with the idea of

Parsing concepts by object types (automatically thru tools and other)

Therefore affording them to know which are the anchors

Therefore affording both participants to know how to unfurl

/ lengthen if needed

Which is by intention

(intention is a not-yet-introduced feature of determining HANDSHAKES)

By intention it is intended to propose Query

Where the proposer and the recipient

Would (perhaps?) able to obtain % analysis

Of the nodes by providing atleast a 50/60%

(actually it depends on culture/ accepting deviations)

(anything below 60% is probably considered (bad) analogy)

.

Anyhow.

This is a critical component of the game

(proposed demo for testing the effectiveness of this)

Because this relates to the ability of the three initial TRANSMUTES

To be able to obtain generalization

.

Obtaining Transmutative rule of this type usually come with queries such as:

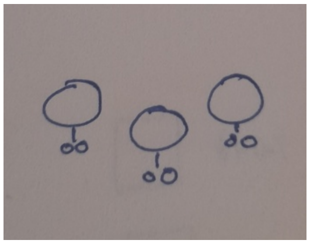
“ Hey claude; do you know any other examples that involve 3 steps like these?”

“ Hey claude; if you are to imply that this is in the field of architecture; how would it look like?”

“ Hey claude; give me the analogy that fits this type of outcome”

T3

pic

|  | Tr3

(3)

| Node1 + Node2 + Node 3 = Node 4 |

| Node1 ~ Node2 ~ Node3 =(~) Node4 |

Onto

| Node[1a.oo.1b.oo.1c] + Node[2a.oo.2b.oo2c] + Node3 [3a.oo.3b] = Node4 |

Transmutative 3

Is intended to obtain the decomposition of each node

When a combination is bracketed | N + N | = N -> |Na.Nb + Nc.Nd| = Ne.Nf

(?)

This is also done in the same spirit of QUANTITY

(likely also because quality is handled elsewhere)

On which there would eventually be derivable math formulas

To refine the data and compare it with a human sampled HANDSHAKE %

To then obtain which is more likely to be which

To then also apply / conjoin with other Transmutes

Some examples include:

“Hey claude; forget about how they connect;

Give me the features of each of the component

And maybe also their treatment in between

“Hey claude; mayhap you can provide me with

All the subcomponents of each of the listed component

In the cars? It will be great if you could

Provide them in separate, independent layer

And also a few decomposition onwards

Thank you!

Some combination later would be able to:

“Hey claude; if we know that

This decomposes to this and this

And that decomposes to that and that

And this is made out of thus and thus

Suppose we learn that one of the this/thus is the same

Are you able to perhaps know the reason why they are similar?

Because they both ###?

Okay this one… might be abit distant. But surely! Some human help or confirmation

Could provide (categorization) ?

Some more notes on tools & format

Selection;

Yepp. we argue that these are (likely?) the more flexible; rather than the transmutes;

so we will deploy or adjust or propose by the deployment;

(Tho hoping to follow up on these analysis later)

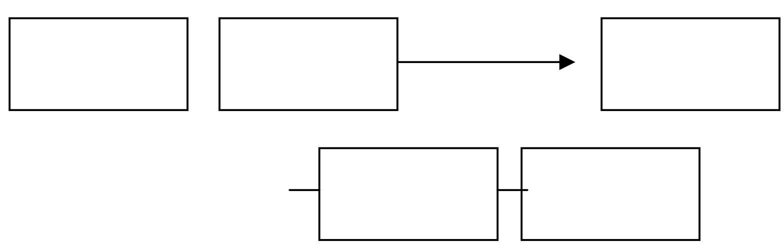
x

Finally, here are some of our proposed

Data structure Flows to design our initial pipeline:

[1] Extracting neurosymbolic representation for objects and transformations

Such like



Sampling data >

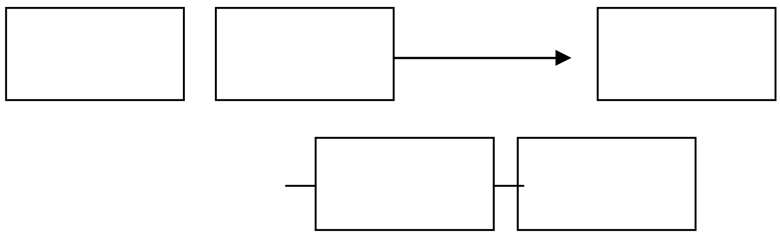
Pre-Process (labeling objects; et) -> Stash

-> LLM Query (labeling) -> Manual Confirmation -> Stash

[2] Extracting / querying 3 selected transmutes

before filtering manually / categorizing after

such like



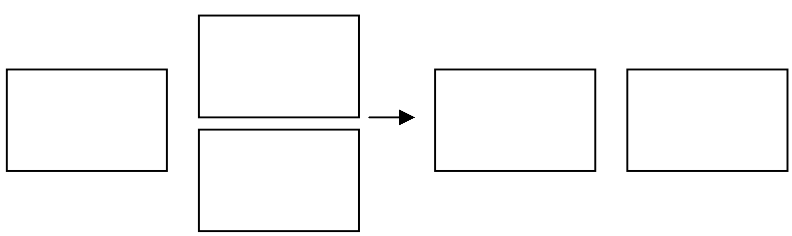
Preparing Data -> Pre Process Data (labeling#2) -> Stash

-> LLM Query for (T1,T2,T3) -> Manual Layer -> Stash

[3] Prompting onto additional groups of

same representation from different layers

or based on proximity to chosen tool



Stash -> Categorizing -> LLM Query / Manual -> Analysis

Gauge proximity; (+ seek to automate)

by decomposing down to tool range and comparing layer amounts

We intend to deploy this to our

Proposal of ## startup

.doc

Thank ye!

REFERENCES

. Loosely inspired by; works on

(yt vids, news)

. On language, geometric deep learning, neurosymbolic language, and conversion / approaches to

mine semantics / decompose text onto symbolic representation

. Chomsky, N. (2021). The minimalist program: 20th anniversary edition. MIT press.

. MIT news: Re-imagining our theories of language (2023)

https://news.mit.edu/2023/re-imagining-our-theories-of-language-0922

. Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. Behavioral and brain sciences, 40.

. Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. Nature Machine Intelligence, 3(6), 472-484.

. Veličković, P., & Bronstein, M. (2021). Graph attention networks. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 15064-15065.

. Friston, K. J., Parr, T., & de Vries, B. (2017). The graphical brain: Belief propagation and active inference. Network Neuroscience, 1(4), 381-414.

. Generating Data for Symbolic Language with Large Language Models Jiacheng Ye♠, Chengzu Li♣, Lingpeng Kong♠, Tao Yu♠

. Zhu, X., Li, Y., Zhang, Y., & Yang, J. (2023). Neural-symbolic reasoning over knowledge graph for multi-modal machine learning. arXiv preprint arXiv:2303.15345.

. Anthropic. (2023). Constitutional AI: A Framework for Machine Learning Systems that Interact with Humans. arXiv preprint arXiv:2310.07749.

. Nye, M., Tessler, M. H., Tenenbaum, J. B., & Lake, B. M. (2023). Learning neuro-symbolic programs for commonsense reasoning. International Conference on Machine Learning (ICML 2023).

. Huang, S., Xu, Y., Xiao, T., Xu, L., Wu, K., & Wang, Z. (2023). Large language models as semantic knowledge optimizers. arXiv preprint arXiv:2309.11206.

. Li, B., Chen, M., Xiao, T., Li, Z., Wang, W., & Wang, X. (2023). Knowledge extraction from large language models. arXiv preprint arXiv:2310.08238.

Et et

-